

QoS em servidores HTTP Apache

José Carlos Martins, Luis Miguel Rato

Resumo—Os serviços baseados na Internet têm registado um crescimento contínuo e acelerado nos últimos anos, dependendo o seu sucesso, em larga medida, da qualidade de serviço (QoS) prestada. A sociedade moderna tornou-se fortemente dependente da Internet e dos vários serviços nela disponibilizados.

Neste artigo é apresentado um sistema original de controlo em cadeia fechada de QoS, aplicado ao servidor HTTP Apache. O objetivo principal do sistema é implementar um mecanismo capaz de realizar a diferenciação de serviço (*DiffServ*) entre duas classes de sítios: *Premium* e *Outros*.

Palavras chave—Servidor HTTP, Apache, Teoria de controlo, QoS, Erro nulo, Rejeição de perturbações, Seguimento de referência.

I. INTRODUÇÃO

COM o advento da "computação em nuvem", os serviços baseados na Internet desempenham um papel cada vez mais importante. O seu sucesso depende, em larga medida, da qualidade de serviço (QoS) prestada[1], o que enfatiza a necessidade de monitorizar a QoS de modo a impor um "Acordo de Nível de Serviço"(ANS)[2].

O aumento do uso de servidores HTTP representa um constante desafio ao seu desenho e desenvolvimento, no sentido de lhes permitir melhorar o desempenho, a disponibilidade e a escalabilidade. Embora na maior parte dos casos não seja técnica e economicamente possível alocar recursos dimensionados para os picos de utilização é, ainda assim, possível fornecer uma melhor QoS a sítios privilegiados (*Premium*), mesmo quando os servidores estão sobrecarregados com pedidos[3].

O recurso a sistemas de QoS é, cada vez mais, uma exigência/tendência no desenho e desenvolvimento de servidores HTTP.

A teoria de controlo é uma ferramenta poderosa no desenho de sistemas de QoS, permitindo alocar de forma dinâmica diferentes recursos a diferentes classes de pedidos baseados em amostras de *feedback*. Pese embora a teoria de controlo seja utilizada em várias áreas, no caso concreto de sistemas computacionais, o objetivo principal é torná-los mais robustos e estáveis. As áreas mais populares de pesquisa têm sido as redes informáticas, sistemas operativos, servidores HTTP, sistemas de gestão de bases de dados (SGBD), multimédia e gestão de energia[4].

Administradores de sistemas experientes sabem que deixar a gestão de sistemas dinâmicos a cargo de operadores não é aceitável, já que as alterações ocorrem tão rapidamente que não é humanamente possível responder em tempo útil[4].

Nas últimas duas décadas foram realizados vários estudos sobre QoS em serviços baseados na Internet, alguns dos quais

com especial ênfase na diferenciação de serviço [4], [5], [6], [7], [8]. Linhas mais recentes de investigação abordam os controladores *fuzzy*, os quais são controladores dinâmicos, não-lineares[9], [10], [11].

II. EQUIPAMENTOS E METODOLOGIA

A. Equipamentos

No sentido de implementar um sistema de QoS aplicado ao servidor HTTP Apache, baseado em teoria de controlo, é implementada uma infraestrutura informática dedicada. Esta infraestrutura é constituída por vários componentes, designadamente: um servidor HTTP Apache modificado (de agora em diante referido como servidor Apache), dois clientes HTTP de agora em diante referidos como clientes), uma *Gateway* (onde reside todo o mecanismo de QoS, baseado em teoria de controlo: controladores PID/transdutor, *etc.*), dois switch e respectiva cablagem.

No quadro seguinte apresentam-se as principais características dos clientes, da *Gateway* e do servidor Apache. Os dois switch utilizados são do padrão 10/100/1000Base-T, com 2Gb/s, por porta, em modo *full duplex*.

Característica	Cientes	Gateway	Servidor Apache
Processador	Intel P8700 2.53GHz	Intel Atom330 1.60GHz	Intel T3400 2.16GHz
Memoria (Gb)	2	2	2
Interface de Rede	10/100/1000Base-T	2 x 10/100/1000Base-T	10/100/1000Base-T
Sistema Operativo	GNU Linux (Fedora 19)	GNU Linux (Fedora 19)	GNU Linux (Fedora 19)
Kernel	3.11.9-200 64bit	3.11.9-200 64bit	2.6.18-371.1.2 64bit

Os vários equipamentos são interligados da seguinte forma:



Fig. 1: Esquema simplificado da infraestrutura: (a) Clientes *Premium* e *Outros*, (b) Switch A, (c) *Gateway*, (d) Switch B, (e) Servidor Apache.

Na figura 2 apresenta-se um esquema simplificado da "ligação física ao sistema de controlo em cadeia fechada". São representados, entre outros aspetos, o fluxo de pacotes, a sua classificação, as filas de espera e o controlador PID.

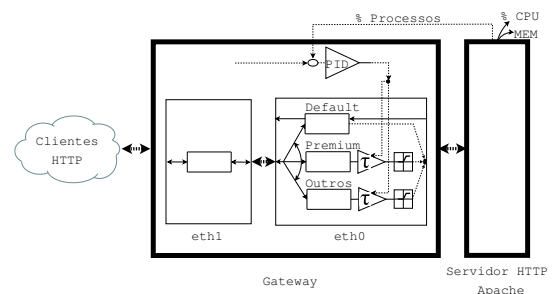


Fig. 2: Esquema simplificado da "ligação física ao sistema de controlo em cadeia fechada aplicado ao servidor HTTP Apache".

B. Otimizações

No sentido de otimizar os recursos disponíveis dos equipamentos e devido ao facto de se poder registar um elevado volume de tráfego de rede e de pedidos HTTP, são realizadas algumas otimizações ao nível dos clientes, da *Gateway* e do servidor Apache.

1) *Clientes HTTP*: Após o arranque dos clientes é executado um *script*, cujo objetivo é otimizar os recursos disponíveis. Em seguida apresenta-se o mesmo.

```
#!/bin/bash
ulimit -n 65535
ulimit -u 2048
ulimit -c unlimited
modprobe tcp_htcp
sysctl -w net.ipv4.tcp_congestion_control=htcp
ifconfig eth0 txqueuelen 10000
```

2) *Servidor Apache*: Neste trabalho é utilizada e modificada a versão 2.2.3 do servidor Apache. O código fonte do servidor Apache é alterado de modo a ser possível recolher informação do seu *Scoreboard*. Essa informação é posteriormente transmitida para a *Gateway*.

É executado um *script* igual ao dos clientes visando os mesmos objetivos.

O servidor Apache tem um endereço IP fixo principal: 192.168.100.100 (interface eth0) e dois endereços fixos virtuais/*alias*: 192.168.100.101, 192.168.100.102/24). Os dois endereços virtuais existem de modo a ser possível criar os *virtual host*. O endereço 192.168.100.101 está associado ao sítio *Premium* enquanto o endereço 192.168.100.102 está associado ao sítio *Outros*. São, igualmente, alterados alguns dos parâmetros do servidor Apache, sendo os mais importantes os seguintes:

```
Timeout 5
KeepAlive On
MaxKeepAliveRequests 20
KeepAliveTimeout 3
<IfModule prefork.c>
StartServers 256
MinSpareServers 5
MaxSpareServers 10
ServerLimit 256
MaxClients 256
MaxRequestsPerChild 0
</IfModule>
```

3) *Gateway*: O mecanismo de QoS implementado na *Gateway* utiliza como principais ferramentas o *tc*, o *HTB* e o *NetEM*. A *Gateway* depois de receber a informação do servidor Apache executa um *script*, cujo objetivo é implementar o mecanismo de QoS, nomeadamente, a classificação do tráfego, definição e aplicação das classes de serviço.

Tal como sucedeu nos clientes e no servidor Apache, após o arranque da *Gateway* é executado o *script* com o objetivo de otimizar os recursos disponíveis. O ficheiro */etc/sysctl.conf* foi igualmente alterado, tendo sido introduzidas as seguintes linhas:

```
net.ipv4.ip_forward = 1
net.ipv4.netfilter.ip_conntrack_max = 32768
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_orphan_retries = 1
net.ipv4.tcp_fin_timeout = 25
net.ipv4.tcp_max_orphans = 8192
net.ipv4.ip_local_port_range = 32768 61000
```

C. Execução e recolha de informação

Os clientes solicitam, através do software *httpperf* (<http://www.hpl.hp.com/research/linux/httpperf/>), ao servidor

Apache a sua página de teste da seguinte forma: `httpperf --hog --server <sítio> - -uri /index.html --wssess=300000,50,2 --burst-len 5 --rate 50 --timeout 5`, em que <sítio> é 192.168.1.101, no caso do sítio *Premium* e 192.168.1.102, no caso do sítio *Outros*.

É escolhida esta página HTML estática, com 5043 bytes, devido ao facto de se pretender que não existam conteúdos dinâmicos, ou acesso a sistemas de gestão de bases de dados (SGBD), os quais poderiam colocar limites/atrasos ao sistema, ou até formar um ponto de congestionamento. Também se pretendeu que a página escolhida não tivesse dimensões muito reduzidas, já que não permitiria gerar tráfego suficiente, nem ser muito grande, o que criaria tráfego em excesso.

III. APRESENTAÇÃO E DISCUSSÃO DE RESULTADOS

A. Sistema sem controlador

O número total de processos Ativos é quase sempre perto dos 100% (figura 3), o que revela a saturação do servidor Apache. Analisando as percentagens dos processos-filho HTTP (de agora em diante referidos como processos), constata-se que não existe diferença significativa entre o sítio *Premium* e o sítio *Outros*. O sistema não faz qualquer diferenciação entre os pedidos do cliente do sítio *Premium* (de agora em diante referido como cliente *Premium*) e os pedidos do cliente do sítio *Outros* (de agora em diante referido como cliente *Outros*). Tendo em consideração a natureza estocástica dos pedidos, existem instantes em que a percentagem de processos do sítio *Outros* é superior à percentagem de processos do sítio *Premium* e vice-versa. Este é o comportamento esperado do sistema, já que o servidor tem capacidade para receber/processar os pedidos dos clientes e está configurado para satisfazer os mesmos conforme vão ocorrendo (FIFO).

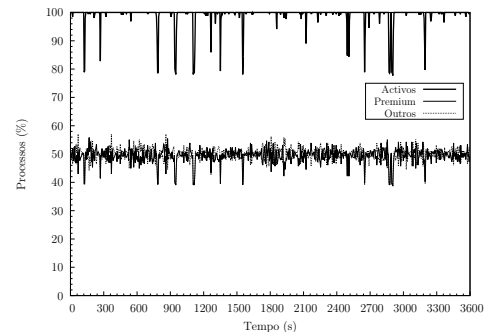


Fig. 3: Sistema de controlo em cadeia fechada sem controlador e com saturação. Servidor Apache: Percentagem absoluta de processos.

B. Sistema com controlador PID

Em seguida vamos analisar o comportamento do sistema de controlo em cadeia fechada com um controlador proporcional-integral-derivativo (PID) e as constantes definidas do seguinte modo: $k_p = 5$, $k_i = 3$, $k_d = 8$. A taxa de sessões é de 50 por segundo (50ss) e cada sessão efectua 50 pedidos. É de salientar que a definição das constantes indicadas resulta de um processo de afinação em que, por exemplo, para valores baixos de k_p (< 2) o sistema não consegue fazer a diferenciação de serviço, já que o cliente *Outros* exhibe percentagens relativas de

processos ativos superiores ao objetivo e, para valores elevados de k_p (> 10) os dois clientes são penalizados e o sistema oscila permanentemente.

1) *Erro nulo*: O objetivo (*setpoint*) definido neste caso é: 80% da capacidade do servidor, expressa em processos *Ativos*, a processar pedidos do sítio *Premium* e os remanescentes 20% a processar pedidos do sítio *Outros*.

A percentagem de processos *Ativos* é muito elevado, estando quase sempre compreendida entre os 90% e os 100% (figura 4). Regista-se uma clara diferenciação entre o cliente *Premium* e o cliente *Outros*.

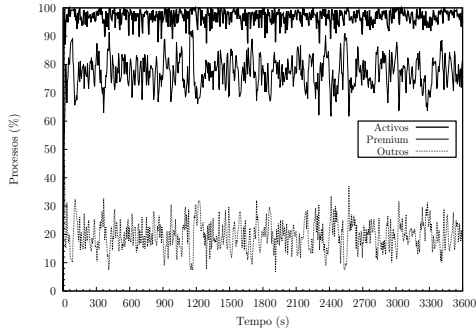


Fig. 4: Sistema de controlo em cadeia fechada com controlador PID. Erro nulo e 50ss. Servidor Apache: Percentagem absoluta de processos.

Na figura 5 constatamos, tal como já era possível observar na figura 4, que o objetivo definido foi atingido. A percentagem relativa de processos a executar pedidos do sítio *Premium* é tendencialmente 80% e do sítio *Outros* é tendencialmente 20%. A natureza estocástica dos pedidos HTTP não permite atingir, na perfeição, o objetivo definido, pelo que este comportamento do sistema não está relacionado com o seu incorreto funcionamento.

As percentagens relativas (PR) de processos são calculadas da seguinte forma:

$$PR_{Proc. \langle Premium|Outros \rangle} (\%) = \frac{Proc. \langle Premium|Outros \rangle (n)}{Proc. Ativos (n)} \cdot 100$$

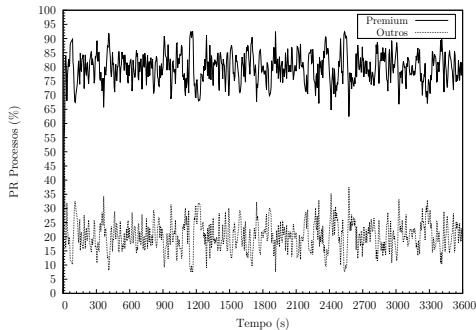


Fig. 5: Sistema de controlo em cadeia fechada com controlador PID. Erro nulo e 50ss. Servidor Apache: Percentagem relativa de processos Ativos.

No que diz respeito ao erro, o mesmo está quase sempre compreendido entre os -10% e os +10%, o que é bom se tivermos em consideração a carga imposta ao servidor e a variabilidade dos pedidos.

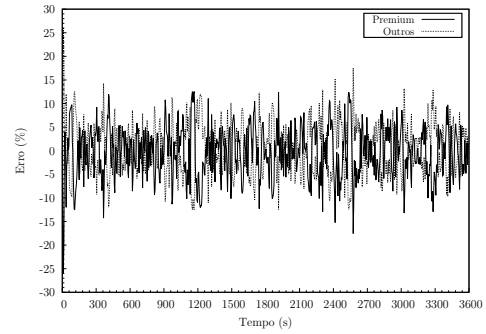


Fig. 6: Sistema de controlo em cadeia fechada com controlador PID. Erro nulo e 50ss. Servidor Apache: Erro.

O sistema consegue ajustar dinamicamente o atraso nos pacotes de modo a atingir o objetivo definido. Tal como esperado, é quase sempre imposto exclusivamente atraso aos pacotes do cliente *Outros* (figura 7). Existem dois momentos (perto dos instantes $t_1 200$ e $t_2 400$) em que também é imposto atraso aos pacotes do cliente *Premium*, mas tal facto não está associado a uma efetiva necessidade de atraso nesses pacotes. Convém mencionar que no código do sistema foram implementadas algumas funcionalidades para testar o sistema face ao *Seguimento de referência* e à *Rejeição de perturbações*. Nestas metodologias o período de observação de 3600s é dividido em três intervalos de 1200s. No final destes intervalos é realizado um *reset* do controlador PID de modo a termos um sistema sem memória/histórico. Esta situação, normalmente indesejada, pode ser encarada como uma perturbação do sistema, mas que atesta a robustez do mesmo, já que ele reage de forma rápida e correta, impondo novamente os atrasos necessários com vista a obtenção dos objetivos traçados.

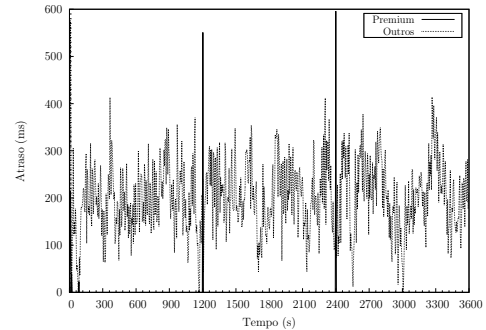


Fig. 7: Sistema de controlo em cadeia fechada com controlador PID. Erro nulo e 50ss. Gateway: Atraso.

A taxa de ligações (lig./s), nos dois clientes, foi semelhante ao sistema sem controlador com saturação. Contudo, no que diz respeito à taxa de pedidos e à taxa média de respostas o sistema exibe um comportamento que permite atestar a ação do controlador, já que nos clientes *Premium* e *Outros* a taxa de pedidos e respostas é inferior ao sistema sem controlador com saturação. É assim perceptível a ação do sistema, em particular sobre o cliente *Outros*, mais concretamente através da imposição de atrasos aos seus pacotes, conforme constatado na figura 7.

Não houve diferença significativa na taxa de ligações entre o cliente *Premium* e o cliente *Outros*: 50,8 e 53,4, respectivamente. No entanto, tendo em consideração o que é referido

anteriormente, já se registou uma diferença expressiva na taxa de pedidos (ped./s): 439,9 no cliente *Premium* e 84,4 no cliente *Outros*. Esta diferença justifica-se se tivermos em consideração os valores obtidos na relação repostas por ligação. No caso do cliente *Premium* este valor foi de 46,963, enquanto no cliente *Outros* foi apenas 16,840. Assim sendo o cliente *Premium* fez, em média, mais pedidos e obteve mais respostas.

Como resultado de uma maior taxa de pedidos do cliente *Premium* e deste ser privilegiado relativamente ao cliente *Outros*, também obteve uma maior taxa média de respostas (resp/s): 426,4 contra 61,8 no cliente *Outros*.

2) *Rejeição de perturbações*: Na figura 8 está representada a percentagem de processos quando existe uma perturbação no instante t_{1200} e a reposição das condições iniciais no instante t_{2400} . A perturbação consiste na introdução de pedidos do cliente *Outros*. O objetivo é verificar como reage o sistema ao aparecimento súbito de uma elevada carga de pedidos do cliente *Outros*, quando este só estava a processar pedidos do cliente *Premium* e subitamente os pedidos do cliente *Outros* desaparecem e o sistema volta a ter exclusivamente pedidos do cliente *Premium*.

Como é natural, nos intervalos t_0 até t_{1200} e perto t_{2400} até t_{3600} existe uma coincidência total entre a percentagem de processos *Ativos* e processos *Premium*, já que o servidor Apache está exclusivamente a processar pedidos do cliente *Premium*. Deve ser salientado que nestes intervalos o sistema tem o controlador ativo, pelo que tenta manter a percentagem relativa de processos *Premium* perto dos 80% (objetivo), mesmo que não existam pedidos do cliente *Outros*. Essa é a razão pela qual o número de processos *Ativos* não se aproxima dos 100%, com exceção no arranque do sistema. Quando começam a surgir pedidos do cliente *Outros*, a percentagem de processos *Ativos* é geralmente superior aos 95%.

É possível constatar que o sistema consegue se adaptar rapidamente após o início dos pedidos do cliente *Outros*, de modo a manter o objetivo traçado: 80% da capacidade do servidor, expressa em processos *Ativos*, a processar pedidos do sítio *Premium* e os remanescentes 20% a processar pedidos do sítio *Outros*. Existe um momento inicial, após a perturbação, em que a percentagem dos processos *Premium* desce até cerca de 40%, mas recupera rapidamente. Esta redução resulta do aumento do atraso imposto aos pacotes destinados ao sítio *Premium*, conforme veremos adiante (figura 10). O aumento do atraso deve-se ao facto da percentagem relativa dos pro-

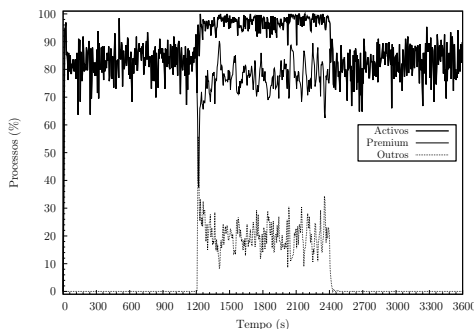


Fig. 8: Sistema de controlo em cadeia fechada com controlador PID. Rejeição de perturbações e 50ss. Servidor Apache: Percentagem absoluta de processos.

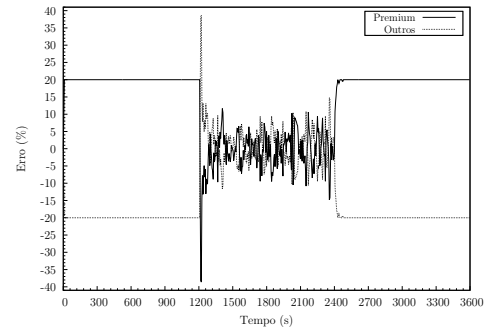


Fig. 9: Sistema de controlo em cadeia fechada com controlador PID. Rejeição de perturbações e 50ss. Servidor Apache: Erro.

cessos *Premium* ser superior aos 80%, o que não é crítico enquanto o servidor só está a processar os seus pedidos, mas o é após existirem pedidos para o sítio *Outros*. O sistema tem de forçar o cumprimento dos objetivos, mesmo que isso implique o aumento temporário do atraso dos pacotes destinados ao sítio *Premium*.

Nos intervalos t_0 a t_{1200} e perto t_{2400} a t_{3600} regista-se um erro de +20% e -20% relativamente ao objetivo do sítios *Premium* e *Outros*, respectivamente. Contudo, este erro não resulta do incorreto funcionamento do sistema mas do facto de não existir, nesses intervalos, pedidos destinados ao sítio *Outros*. Nos segundos imediatamente após o instante t_{1200} regista-se um erro positivo dos processos do sítio *Outros* e negativo dos processos do sítio *Premium*. Este erro resulta da ação do sistema ao detetar que, no instante t_{1200} , existe uma percentagem relativa superior à definida para os processos do sítio *Premium* e uma percentagem relativa inferior à definida para os processos do sítio *Outros*, o que força o sistema a impor um atraso aos pacotes destinados ao sítio *Premium* nesse instante (figura 10).

Se atentarmos ao atraso imposto aos pacotes verificamos que:

- Até ao instante t_{1200} é imposto um atraso de cerca de 400ms aos pacotes destinados ao sítio *Premium* (figura 10). O intuito do sistema é forçar o cumprimento do objetivo de 80% de processos HTTP *Ativos* a processar pedidos do sítio *Premium* e impedir que os mesmos “monopolizem” o servidor. Caso se registasse essa saturação com pedidos *Premium*, o sistema podia levar muito tempo a estabilizar após o aparecimento dos primeiros pedidos do cliente *Outros*. Conforme foi constatado anteriormente, o sistema consegue atingir os objetivos definidos muito rapidamente;
- No instante t_{1200} é imposto um atraso de quase 1000ms aos pacotes destinados ao sítio *Premium*, devido às razões explicadas anteriormente;
- Entre o instante t_{1200} e o instante t_{2400} é unicamente imposto limite aos pacotes destinados ao sítio *Outros*, sendo o atraso variável. Mais uma vez se constata o comportamento dinâmico do sistema;
- Após o instante t_{2400} e devido ao facto de não existirem novamente pedidos destinados ao sítio *Outros*, só são impostos atrasos aos pacotes destinados ao sítio *Premium*, sendo este atraso maior no instante t_{2400} . Este maior

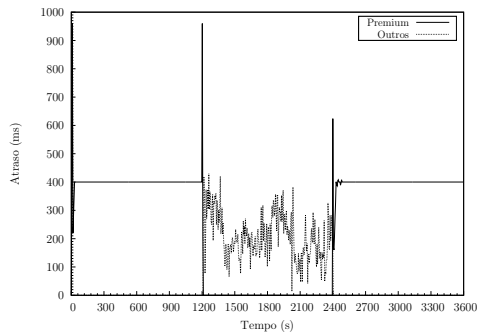


Fig. 10: Sistema de controle em cadeia fechada com controlador PID. Rejeição de perturbações e 50ss. Gateway: Atraso.

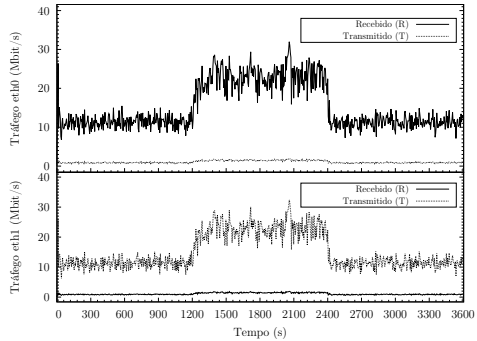


Fig. 11: Sistema de controle em cadeia fechada com controlador PID. Rejeição de perturbações e 50ss. Gateway: Tráfego de rede.

atraso permite ao sistema atingir mais rapidamente o objetivo definido.

Tal como esperado, no que diz respeito ao tráfego de rede, existem três zonas distintas (figura 11), as quais coincidem com os intervalos em que só existem pedidos destinados ao sítio *Premium* (instante t_0 a instante t_{1200} e instante t_{2400} a instante t_{3600}) e o intervalo em que os pedidos destinados aos dois sítios coexistem (instante t_{1200} a instante t_{2400}). Como é lógico, neste sistema, o tráfego de rede, salvo algum fator externo (que não se regista) resulta basicamente do volume de informação a transferir dos pedidos dos clientes e das respostas do servidor. Assim sendo, é natural que o tráfego com os dois clientes ativos se traduza em mais pedidos e respostas, o que resulta num aumento do tráfego de rede.

No caso do cliente *Premium*, não é possível fazer uma análise comparativa relativamente às observações anteriores já que assentam em pressupostos distintos, nomeadamente a coexistência permanente de pedidos dos dois clientes. No que diz respeito ao cliente *Outros* é possível fazer essa análise, já que no período em que efectuou pedidos, os mesmos coexistem com os pedidos do cliente *Premium* e com o mesmo objetivo. A taxa de ligações (lig/s) do cliente *Outros* é muito semelhante ao *Erro nulo*: 53,3 e 53,4, respectivamente. Também a taxa de pedidos (pedidos/s) é bastante semelhante: 86,1 contra 84,4. O mesmo sucede para a taxa média de respostas (respostas/s): 64,2 contra 61,8 resp/s. É, assim, validado o comportamento do sistema durante o período em que os pedidos coexistiam.

3) *Seguimento de Referência*: Neste caso, o objetivo é variável, isto é: em determinados intervalos o objetivo é 80% da capacidade do servidor, expressa em processos *Ativos*, a processar pedidos do sítio *Premium* e os remanescentes 20%

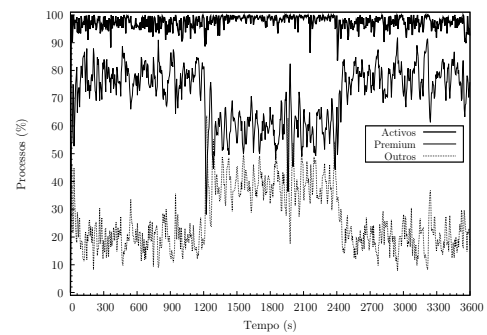


Fig. 12: Sistema de controle em cadeia fechada com controlador PID. Seguimento de referência e 50ss. Servidor Apache: Percentagem absoluta de processos.

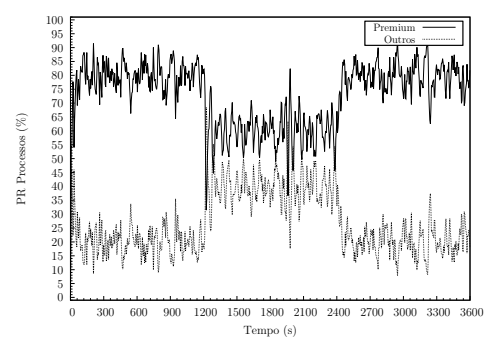


Fig. 13: Sistema de controle em cadeia fechada com controlador PID. Seguimento de referência e 50ss. Servidor Apache: Percentagem relativa de processos Ativos.

a processar pedidos do sítio *Outros* (instante t_0 a instante t_{1200} ; instante t_{2400} a instante t_{3600}) e noutro intervalo o objetivo é 60%/40% (instante t_{1200} a instante t_{2400}).

Tal como em casos anteriores, constata-se a tendência para a saturação do servidor Apache. A percentagem de processos HTTP *Ativos* está quase sempre compreendida entre os 90% e os 100%. Tal como na *Rejeição de perturbações*, é possível observar três zonas distintas, estando as mesmas relacionadas com o objetivo definido: 80%/20% e 60%/40%.

É possível notar que, mais uma vez, o sistema funcionou como pretendido, ou seja: cumpre os objetivos definidos, é rápido a reagir às alterações (Figura 12) e é estável. À semelhança da *Rejeição de perturbações*, perto do instante t_{1200} as percentagens denotam grande diferença relativamente ao pretendido mas, mais uma vez, corresponde ao instante em que o sistema se está a adaptar à alteração dos objetivos, mais concretamente à redução das percentagens relativas do cliente *Premium* e aumento das percentagens relativas do cliente *Outros*.

De um modo geral, o erro das percentagens dos processos está compreendido num intervalo aceitável: -15% e +15%. Existem quatro exceções:

- Instante t_{1200} : Adaptação do sistema aos novos objetivos 60%/40%;
- Perto do instante t_{2000} : Sem razão específica. Resulta da natureza estocástica dos pedidos HTTP;
- Instante t_{2400} : Adaptação do sistema aos novos objetivos 80%/20%;
- Perto do instante t_{3200} : Sem razão específica. Resulta da natureza estocástica dos pedidos HTTP.

Conforme se pode comprovar na figura 14, entre o instante

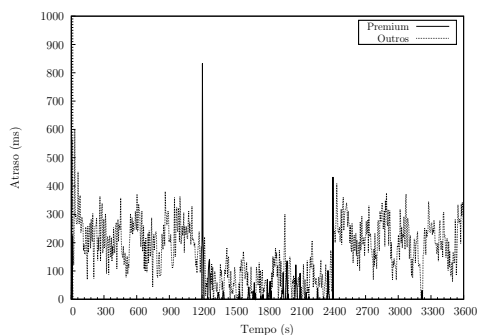


Fig. 14: Sistema de controlo em cadeia fechada com controlador PID. Seguimento de referência e 50ss. Gateway: Atraso.

t_{1200} e o instante t_{2400} são aplicados pontualmente pequenos atrasos aos pacotes destinados ao sítio *Premium*, facto que não sucede fora deste intervalo. Esta situação pode ser explicada, pelo facto do objetivo dos processos *Premium* não diferir muito do objetivo dos processos *Outros*, 60% e 40% respectivamente. Devido à natureza variável dos pedidos, bem como ao facto de uma sessão poder fazer mais de um pedido (o que ocupa um processo durante um período de tempo não determinado), o sistema necessitou de aplicar atrasos aos pacotes destinados aos dois sítios.

No que diz respeito ao tráfego de rede, e ao contrário do que sucedeu com no caso da *Rejeição de perturbações*, não se registam zonas distintas no período de observação. O mesmo está compreendido entre os 10Mbit/s e os 35Mbit/s, com maior frequência entre os 15Mbit/s e os 30Mbit/s, pelo que a alteração dos objetivos durante o período de observação não teve influência neste aspeto.

O cliente *Outros* tem uma taxa de ligações superior ao cliente *Premium* (104,1%), mas apresenta uma taxa de pedidos e uma taxa média de repostas bem inferior, 30,3% e 26,1%, respectivamente.

Por fim convém salientar que nem o CPU nem a memória foram fatores limitativos nos resultados apresentados, já que em todos os casos durante o período de observação:

- O processador do servidor Apache esteve sempre mais de 80% *idle*;
- A memória utilizada do servidor Apache foi sempre muito estável, rondando os 512 MB o que representa cerca de 25% da capacidade do servidor (2 GB). É de mencionar que na configuração do servidor HTTP Apache o parâmetro `StartServers` é igual ao parâmetro `MaxClients` e `ServerLimit`: 256, pelo que o servidor inicia a sua atividade logo com a capacidade máxima de processos;
- O tráfego de rede nas interfaces da *Gateway* foi sempre inferior a 60 MB/s, bastante inferior à sua capacidade: 1 Gb/s.

O programa *httpperf* reportou que todas as repostas foram do tipo 2xx, ou seja: “Sucesso”, o que indica que a ação foi recebida com sucesso, compreendida e aceite. Nunca se registaram erros devido à falta de descritores de ficheiros, falta de portas TCP disponíveis ou exaustão da tabela de descritores de ficheiros do sistema. Este aspeto é muito relevante, pois caso algum destes contadores seja diferente de zero, revela

incapacidade do cliente em criar e manter a carga de pedidos ao servidor HTTP Apache, o que inviabiliza os resultados. Não existem, igualmente, outro tipo de erros, o que justificaria avaliar a sua razão.

IV. CONCLUSÕES

Neste trabalho é definido e implementado um sistema original de controlo em cadeia fechada de QoS, aplicado ao servidor HTTP Apache. Este sistema implementa uma ferramenta robusta e estável de recolha e análise de informação, bem como a gestão de recursos dinâmicos. Por fim, é comprovada sua capacidade de realizar a diferenciação de serviço pretendida entre as duas classes de sítios: *Premium* e *Outros*. Esta qualidade é comprovada para o *Erro nulo*, *Rejeição de perturbações* e *Seguimento de referência com erro nulo*.

ANEXO A

Parâmetro	Sistema sem controlador	
	Premium	Outros
Taxa de ligações (a)	50,2	50,2
Taxa de pedidos (b)	488,7	490,1
Taxa média de repostas (c)	487,4	489,2
Respostas por ligação (d)	9,956	9,957

(a) lig./s, (b) ped./s, (c) resp./s, (d) resp./lig.

Parâmetro	Sistema com controlador PID					
	Erro nulo		Rej. Pertub.		Seg. Ref.	
	Premium	Outros	Premium	Outros	Premium	Outros
Taxa de ligações (a)	50,8	53,4	66,8	53,3	50,8	52,9
Taxa de pedidos (b)	439,9	84,4	338,4	86,1	403,3	122,3
Taxa média de repostas (c)	426,4	61,8	283,4	64,2	389,9	101,6
Respostas por ligação (d)	46,963	16,840	13,801	17,682	46,677	25,509

(a) lig./s, (b) ped./s, (c) resp./s, (d) resp./lig.

BIBLIOGRAFIA

- [1] X. Huang, “UsageQoS: Estimating the QoS of Web services through online user communities,” *ACM Transactions on the Web (TWEB)*, vol. 8, no. 1, p. 1, 2013.
- [2] M. A. Serhani, Y. Atif, and A. Benharref, “Towards an Adaptive QoS-driven Monitoring of Cloud SaaS,” *Int. J. Grid Util. Comput.*, vol. 5, no. 4, p. 263–277, 2014.
- [3] K. H. Chan and X. Chu, “Using Fuzzy PI Controller to Provide QoS on Web Servers,” *Jornal*, 2007.
- [4] J. Hellerstein *et al.*, *Feedback Control of Computing Systems*. John Wiley & Sons IEE Press, 2004.
- [5] M. D. A. M. J. Almedia, P. Cao, J. A. M. D. A. Manikutty, and P. Cao, “Providing differentiated levels of service in web content hosting,” in *Proc. First Workshop Internet Server Performance*, 1998.
- [6] L. Eggert and J. Heidemann, “Application-level differentiated services for Web servers,” *World Wide Web*, vol. 2, no. 3, p. 133–142, 1999.
- [7] T. F. Abdelzaher and K. G. Shin, “Qos provisioning with qcontracts in web and multimedia servers,” in *Real-Time Systems Symposium, 1999. Proceedings. The 20th IEEE*. IEEE, 1999, p. 44–53.
- [8] J. M. Blanquer *et al.*, “QoS for internet services: done right,” in *Proceedings of the 11th workshop on ACM SIGOPS European workshop*. ACM, 2004, p. 8.
- [9] M. Loudini and S. Rezig, “Enhancing Web Server Relative Delay Services by an Integrated SA-fuzzy Logic Controller,” *Int. J. Web Eng. Technol.*, vol. 8, no. 1, p. 27–57, Mar. 2013.
- [10] K. H. Chan and X. Chu, “Design of a Fuzzy PI Controller to Guarantee Proportional Delay Differentiation on Web Servers,” in *Proceedings of the 3rd International Conference on Algorithmic Aspects in Information and Management*, ser. AAIM '07. Springer-Verlag, 2007, p. 389–398.
- [11] P. Pivoňka, “Comparative analysis of fuzzy PI/PD/PID controller based on classical PID controller approach,” in *Proceedings of the 2002 IEEE International Conference on Fuzzy systems*, 2002, p. 541–546.